

Cover Page

Milestone #: 2

Date: 10/20/23

Group Number: 62

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|---------------|-----------------------|--------------------------|---------------------------------|
| Samuel Kim | 71186696 | j2g8k | samuelkim019@gmail.com |
| Pranjal Gupta | 61579363 | y1v1c | pranjalgupta2802@gmail.com |
| Rafael Park | 84277318 | p8w2e | parkrafael11@gmail.com |

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Summary of Project:

Our project will authenticate a user by their CWL and we maintain their social connections (following/followers) and their user-provided data, such as reviews, ratings, images, and videos of their favorite (or least favorite) restaurants they have visited. Simply put, our app will allow users to create and view food review posts.

Changes to ER Diagram:

userAccount:

- User now has 4 attributes as suggested by TA
- Added "Email" and "Password" attributes, added "ProfileID" key
- Changed name from "User" to "userAccount"

userProfile:

- Changed name from "Profile" to "userProfile"

PhotoAlbumPost & VideoAlbumPost:

- Removed both entities, made a direct relation from "Post" to "Photo" and "Video"
- Added primary keys to "Photo" and "Video" with recommendation from TA
- Added constraints to "Photo" and "Video"

Comments & Tags:

- We established a many to one relationship from "Comments" & "Tags" to "Post"
- Added a corresponding attribute for each entity (ie. "Comment" and "Tag")

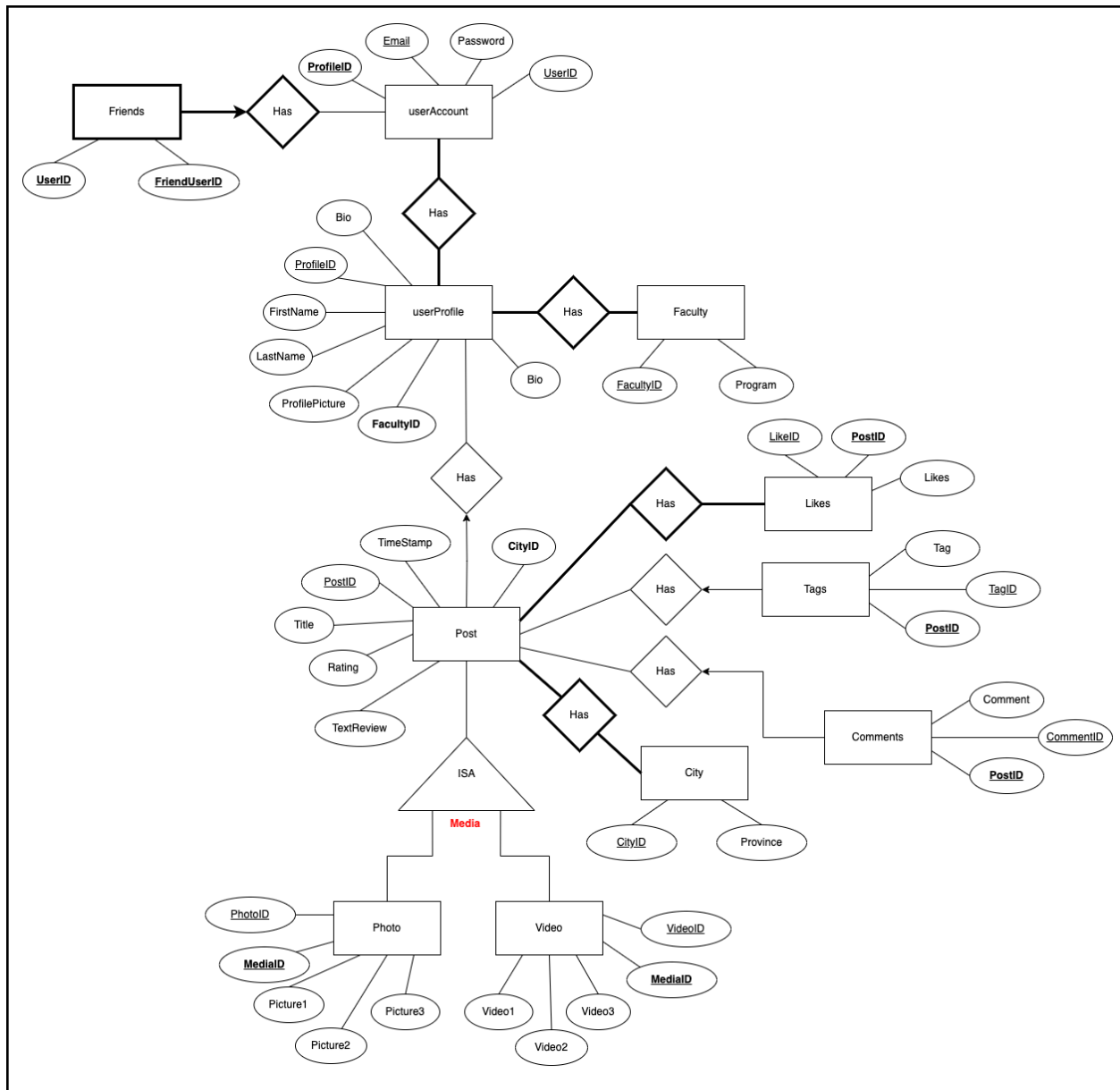
Profile:

- Divided "Name" into two attributes "FirstName" & "LastName"

Messages:

- Deleted Entity

ER Diagram:



ER Diagram Schema:

| <u>Entity</u> | <u>Primary Key</u> | <u>Foreign Key</u> |
|---|---|---------------------------------|
| userAccount (ProfileID : Char, Email : Char, Password : String, UserID : Char) | Primary Keys - UserID , ProfileID , Email | Foreign Keys - ProfileID |
| userProfile (ProfileID : Char, FirstName : Char, LastName : Char, ProfilePicture : image(PNG, JPEG), FacultyID : Char, Bio : String) | Primary Keys - ProfileID | Foreign Keys - FacultyID |
| Faculty (FacultyID : Char, Program : Char) | Primary Keys - Faculty ID | Foreign Keys - N/A |
| Media (MediaID : Int, PostID : Char) | Primary Keys - MediaID | Foreign Keys - PostID |
| Photo ISA Media (PhotoID : Int , Photo1 : image (PNG, JPEG), Photo2 : image (PNG, JPEG), Photo3 : image (PNG, JPEG)) | Primary Keys - VideoID | Foreign Keys - N/A |
| Post (PostID : Int, Title : String, Rating : Int, TextReview : String, TimeStamp : Char, CityID : Char) | Primary Keys - Post ID | Foreign Keys - CityID |

| | | |
|--|---|---|
| Video <i>ISA Media</i> (VideoID: Integer, Video1: MP4, Video2: MP4, Video3: MP4) | Primary Keys - VideoID | Foreign Keys - N/A |
| City (CityID: Char, Province: Char) | Primary Keys - CityID | Foreign Keys - N/A |
| Friends (FriendUserID: Char, UserID: Char) | Primary Keys - FriendUserID, UserID | Foreign Keys - FriendUserID, UserID |
| Likes (LikeID: Char, PostID: Char, Likes : Int) | Primary Keys - LikeID, PostID | Foreign Keys - PostID |
| Tags (TagID : Char, PostID : Char, Tags : Int) | Primary Keys - TagID, PostID | Foreign Keys - PostID |
| Comments (CommentsID : Char, PostID : Char , Comments : String) | Primary Keys - CommentsID, PostID | Foreign Keys - PostID |

Functional Dependencies:

Photo:

PhotoID -> PhotoID, Photo1, Photo2, Photo3

Video:

VideoID -> VideoID, Video1, Video2, Video3

Media:

PostID -> PostID, MediaID

City:

CityID -> CityID, Province

Friends:

FriendUserID -> FriendUserID UserID

User:

UserID, Email, ProfileID -> UserID, Email, ProfileID, Password

Profile:

Profile ID -> Profile ID, FirstName , LastName , ProfilePicture , Bio , Faculty

Faculty -> Program

Faculty:

FacultyID -> FacultyID, Program

Likes:

LikeID -> LikeID, PostID, Likes

Tags:

TagsID -> TagsID, PostID, Tags

Comments:

CommentsID -> CommentsID, PostID, Comments

Post:

PostID -> PostID, Title, Rating, textReview, TimeStamp , CityID

CityID -> Province

Constraints:

For ISA:

- Photos are of type media but can only accept PNG and JPEG.
- Videos are a type of media but can only accept .mp4
- Email needs to be valid email addresses (ie [...]@[email].com)
- Faculty provided by the user should be valid.

Normalization:

Decomposition For Post

Post(PostID, Title, Rating, textReview, TimeStamp, Province, CityID)

FD - (PostID → PostID, Title, Rating, textReview, TimeStamp, CityID)

FD - (CityID → Province)

Closure of PostID (Candidate Key) - (PostID, Title, Rating, textReview, TimeStamp, CityID, Province)

FD - (CityID → Province) violates the 3NF condition

Decomposing to 3NF -

R1 - (CityID, Province)

R2 - PostID - (PostID, Title, Rating, textReview, TimeStamp, CityID)

Decomposition For userProfile

Profile - (Profile ID, FirstName, LastName, ProfilePicture, Bio, Faculty, Program)

FD - (Profile ID → Profile ID, FirstName, LastName, ProfilePicture, Bio, Faculty)

FD - (Faculty → Program)

Closure of Profile ID (Candidate Key) - (Profile ID, FirstName, LastName, ProfilePicture, Bio, Faculty, Program)

FD - (Faculty → Program) Violates the 3NF Condition

Decomposing to 3NF -

R1 - (Faculty, Program)

R2 - (Profile ID, FirstName, LastName, ProfilePicture, Bio, Faculty)

SQL DDL Statements:

-- Creating the Faculty table

```
CREATE TABLE Faculty (  
    FacultyID CHAR(4) NOT NULL,  
    Program VARCHAR2(255) NOT NULL,  
    PRIMARY KEY (FacultyID)  
);
```

-- Creating the City table

```
CREATE TABLE City (  
    CityID CHAR(4) NOT NULL,  
    City VARCHAR2(50) NOT NULL,  
    Province VARCHAR2(20) NOT NULL,  
    PRIMARY KEY (CityID)  
);
```

-- Creating the userProfile table

```
CREATE TABLE userProfile (  
    ProfileID CHAR(4) NOT NULL,  
    FirstName VARCHAR2(50) NOT NULL,  
    LastName VARCHAR2(50) NOT NULL,  
    ProfilePicture BLOB,  
    Bio VARCHAR2(255),  
    Faculty CHAR(4),  
    PRIMARY KEY (ProfileID),  
    FOREIGN KEY (Faculty) REFERENCES Faculty(FacultyID)  
);
```

-- Modifying the userAccount table

```
CREATE TABLE userAccount (  
    UserID CHAR(4) NOT NULL,  
    Email VARCHAR2(100) NOT NULL UNIQUE,  
    ProfileID CHAR(4) NOT NULL,  
    Password VARCHAR2(255) NOT NULL,  
    PRIMARY KEY (UserID, Email, ProfileID),  
    FOREIGN KEY (ProfileID) REFERENCES userProfile(ProfileID)  
);
```

-- Modifying the Post table to associate with a Profile

```
CREATE TABLE Post (  
    PostID NUMBER(10) NOT NULL,  
    Title VARCHAR2(255) NOT NULL,  
    Rating NUMBER(10),  
    TextReview VARCHAR2(255),  
    City CHAR(4) NOT NULL,  
    Timestamp VARCHAR2(23),  
    ProfileID CHAR(4) NOT NULL,  
    PRIMARY KEY (PostID),  
    FOREIGN KEY (City) REFERENCES City(CityID),  
    FOREIGN KEY (ProfileID) REFERENCES userProfile(ProfileID)  
);
```

-- Creating the Media table

```
CREATE TABLE Media (  
    MediaID NUMBER(10) NOT NULL,  
    PostID NUMBER(10) NOT NULL,  
    PRIMARY KEY (MediaID, PostID),  
    FOREIGN KEY (PostID) REFERENCES Post(PostID)  
);
```

-- Modifying the Photo table

```
CREATE TABLE Photo (  
    PhotoID NUMBER(10) NOT NULL,  
    MediaID NUMBER(10) NOT NULL,  
    MediaPostID NUMBER(10) NOT NULL,  
    Photo1 BLOB,  
    Photo2 BLOB,  
    Photo3 BLOB,  
    PRIMARY KEY (PhotoID),  
    FOREIGN KEY (MediaID, MediaPostID) REFERENCES Media(MediaID, PostID)  
);
```

-- Modifying the Video table

```
CREATE TABLE Video (  
    VideoID NUMBER(10) NOT NULL,  
    MediaID NUMBER(10) NOT NULL,  
    MediaPostID NUMBER(10) NOT NULL,  
    Video1 BLOB,  
    Video2 BLOB,  
    Video3 BLOB,  
    PRIMARY KEY (VideoID),  
    FOREIGN KEY (MediaID, MediaPostID) REFERENCES Media(MediaID, PostID)  
);
```

-- For Likes

```
CREATE TABLE Likes (  
    LikeID CHAR(4) NOT NULL,
```

```
PostID NUMBER(10) UNIQUE,  
Likes NUMBER(10),  
PRIMARY KEY (LikeID),  
FOREIGN KEY (PostID) REFERENCES Post(PostID)  
);
```

```
-- For Tags  
CREATE TABLE Tags (  
    TagsID CHAR(4) NOT NULL,  
    PostID NUMBER(10),  
    UserID CHAR(4) NOT NULL,  
    Email VARCHAR2(100) NOT NULL,  
    ProfileID CHAR(4) NOT NULL,  
    TagContent VARCHAR2(255),  
    PRIMARY KEY (TagsID),  
    FOREIGN KEY (PostID) REFERENCES Post(PostID),  
    FOREIGN KEY (UserID, Email, ProfileID) REFERENCES userAccount(UserID, Email, ProfileID)  
);
```

```
-- For Comments  
CREATE TABLE Comments (  
    CommentsID CHAR(4) NOT NULL,  
    PostID NUMBER(10),  
    UserID CHAR(4) NOT NULL,  
    Email VARCHAR2(100) NOT NULL,  
    ProfileID CHAR(4) NOT NULL,  
    CommentContent VARCHAR2(500),  
    PRIMARY KEY (CommentsID),  
    FOREIGN KEY (PostID) REFERENCES Post(PostID),  
    FOREIGN KEY (UserID, Email, ProfileID) REFERENCES userAccount(UserID, Email, ProfileID)  
);
```

```
-- Creating the Friends table
CREATE TABLE Friends (
  UserID CHAR(4) NOT NULL,
  UserEmail VARCHAR2(100) NOT NULL,
  UserProfileID CHAR(4) NOT NULL,
  FriendUserID CHAR(4) NOT NULL,
  FriendUserEmail VARCHAR2(100) NOT NULL,
  FriendUserProfileID CHAR(4) NOT NULL,
  PRIMARY KEY (UserID, UserEmail, UserProfileID, FriendUserID, FriendUserEmail,
FriendUserProfileID),
  FOREIGN KEY (UserID, UserEmail, UserProfileID) REFERENCES userAccount(UserID, Email,
ProfileID),
  FOREIGN KEY (FriendUserID, FriendUserEmail, FriendUserProfileID) REFERENCES
userAccount(UserID, Email, ProfileID)
);
```

Insert Statements:

-- Inserting data into Faculty

```
INSERT INTO Faculty (FacultyID, Program) VALUES ('F001', 'Engineering');
INSERT INTO Faculty (FacultyID, Program) VALUES ('F002', 'Arts');
INSERT INTO Faculty (FacultyID, Program) VALUES ('F003', 'Science');
INSERT INTO Faculty (FacultyID, Program) VALUES ('F004', 'Business');
INSERT INTO Faculty (FacultyID, Program) VALUES ('F005', 'Law');
```

-- Inserting data into City

```
INSERT INTO City (CityID, City, Province) VALUES ('C001', 'Vancouver', 'BC');
INSERT INTO City (CityID, City, Province) VALUES ('C002', 'Toronto', 'ON');
INSERT INTO City (CityID, City, Province) VALUES ('C003', 'Calgary', 'AB');
INSERT INTO City (CityID, City, Province) VALUES ('C004', 'Montreal', 'QC');
INSERT INTO City (CityID, City, Province) VALUES ('C005', 'Halifax', 'NS');
```

-- Inserting data into userProfile

```
INSERT INTO userProfile (ProfileID, FirstName, LastName, Bio, Faculty) VALUES ('P001', 'Phoebe',
'Bridgers', 'Singer and Songwriter', 'F002');
INSERT INTO userProfile (ProfileID, FirstName, LastName, Bio, Faculty) VALUES ('P002', 'Taylor', 'Swift',
'Singer and Songwriter', 'F002');
INSERT INTO userProfile (ProfileID, FirstName, LastName, Bio, Faculty) VALUES ('P003', 'Billie', 'Eilish',
'Singer and Songwriter', 'F002');
INSERT INTO userProfile (ProfileID, FirstName, LastName, Bio, Faculty) VALUES ('P004', 'Frank',
'Ocean', 'Singer and Songwriter', 'F002');
INSERT INTO userProfile (ProfileID, FirstName, LastName, Bio, Faculty) VALUES ('P005', 'Bon', 'Iver',
'Singer and Songwriter', 'F002');
```

-- Inserting data into userAccount

```
INSERT INTO userAccount (UserID, Email, ProfileID, Password) VALUES ('U001', 'phoebe@email.com',
'P001', 'phoebe123');
INSERT INTO userAccount (UserID, Email, ProfileID, Password) VALUES ('U002', 'taylor@email.com',
'P002', 'taylor123');
INSERT INTO userAccount (UserID, Email, ProfileID, Password) VALUES ('U003', 'billie@email.com',
'P003', 'billie123');
INSERT INTO userAccount (UserID, Email, ProfileID, Password) VALUES ('U004', 'frank@email.com',
'P004', 'frank123');
INSERT INTO userAccount (UserID, Email, ProfileID, Password) VALUES ('U005', 'bon@email.com',
'P005', 'bon123');
```

-- Inserting data into Post

```
INSERT INTO Post (PostID, Title, Rating, TextReview, City, Timestamp, ProfileID) VALUES (1, 'Concert Review', 5, 'Amazing Performance!', 'C001', '01-JAN-2023 10:00:00 AM', 'P001');
```

```
INSERT INTO Post (PostID, Title, Rating, TextReview, City, Timestamp, ProfileID) VALUES (2, 'Album Review', 4, 'Loved the tracks!', 'C002', '02-JAN-2023 11:00:00 AM', 'P002');
```

```
INSERT INTO Post (PostID, Title, Rating, TextReview, City, Timestamp, ProfileID) VALUES (3, 'Single Track Review', 5, 'Song on repeat!', 'C003', '03-JAN-2023 01:00:00 PM', 'P003');
```

```
INSERT INTO Post (PostID, Title, Rating, TextReview, City, Timestamp, ProfileID) VALUES (4, 'Tour Review', 4, 'Great energy!', 'C004', '04-JAN-2023 02:30:00 PM', 'P004');
```

```
INSERT INTO Post (PostID, Title, Rating, TextReview, City, Timestamp, ProfileID) VALUES (5, 'Live Session Review', 5, 'Felt the vibes!', 'C005', '05-JAN-2023 03:45:00 PM', 'P005');
```

-- Inserting data into Media

```
INSERT INTO Media (MediaID, PostID) VALUES (1, 1);
```

```
INSERT INTO Media (MediaID, PostID) VALUES (2, 2);
```

```
INSERT INTO Media (MediaID, PostID) VALUES (3, 3);
```

```
INSERT INTO Media (MediaID, PostID) VALUES (4, 4);
```

```
INSERT INTO Media (MediaID, PostID) VALUES (5, 5);
```

-- Inserting data into Photo

```
INSERT INTO Photo (PhotoID, MediaID, MediaPostID) VALUES (1, 1, 1);
```

```
INSERT INTO Photo (PhotoID, MediaID, MediaPostID) VALUES (2, 2, 2);
```

```
INSERT INTO Photo (PhotoID, MediaID, MediaPostID) VALUES (3, 3, 3);
```

```
INSERT INTO Photo (PhotoID, MediaID, MediaPostID) VALUES (4, 4, 4);
```

```
INSERT INTO Photo (PhotoID, MediaID, MediaPostID) VALUES (5, 5, 5);
```

-- Inserting data into Video

```
INSERT INTO Video (VideoID, MediaID, MediaPostID) VALUES (6, 1, 1);
```

```
INSERT INTO Video (VideoID, MediaID, MediaPostID) VALUES (7, 2, 2);
```

```
INSERT INTO Video (VideoID, MediaID, MediaPostID) VALUES (8, 3, 3);
```

```
INSERT INTO Video (VideoID, MediaID, MediaPostID) VALUES (9, 4, 4);
```

```
INSERT INTO Video (VideoID, MediaID, MediaPostID) VALUES (10, 5, 5);
```

-- Likes

```
INSERT INTO Likes (LikeID, PostID, Likes) VALUES ('L001', 1, 5);
```

```
INSERT INTO Likes (LikeID, PostID, Likes) VALUES ('L002', 2, 10);
```

```
INSERT INTO Likes (LikeID, PostID, Likes) VALUES ('L003', 3, 15);
```

```
INSERT INTO Likes (LikeID, PostID, Likes) VALUES ('L004', 4, 20);
```

```
INSERT INTO Likes (LikeID, PostID, Likes) VALUES ('L005', 5, 25);
```

-- Inserting data into Tags

```
INSERT INTO Tags (TagsID, PostID, UserID, Email, ProfileID, TagContent) VALUES ('T001', 1, 'U001', 'phoebe@email.com', 'P001', '#concert');
```

```
INSERT INTO Tags (TagsID, PostID, UserID, Email, ProfileID, TagContent) VALUES ('T002', 2, 'U002', 'taylor@email.com', 'P002', '#album');
```

```
INSERT INTO Tags (TagsID, PostID, UserID, Email, ProfileID, TagContent) VALUES ('T003', 3, 'U003', 'billie@email.com', 'P003', '#song');
```

```
INSERT INTO Tags (TagsID, PostID, UserID, Email, ProfileID, TagContent) VALUES ('T004', 4, 'U004', 'frank@email.com', 'P004', '#tour');
```

```
INSERT INTO Tags (TagsID, PostID, UserID, Email, ProfileID, TagContent) VALUES ('T005', 5, 'U005', 'bon@email.com', 'P005', '#live');
```

-- Inserting data into Comments

```
INSERT INTO Comments (CommentsID, PostID, UserID, Email, ProfileID, CommentContent) VALUES ('C001', 1, 'U001', 'phoebe@email.com', 'P001', 'Amazing concert!');
```

```
INSERT INTO Comments (CommentsID, PostID, UserID, Email, ProfileID, CommentContent) VALUES ('C002', 2, 'U002', 'taylor@email.com', 'P002', 'This album is a masterpiece.');
```

```
INSERT INTO Comments (CommentsID, PostID, UserID, Email, ProfileID, CommentContent) VALUES ('C003', 3, 'U003', 'billie@email.com', 'P003', 'Great song!');
```

```
INSERT INTO Comments (CommentsID, PostID, UserID, Email, ProfileID, CommentContent) VALUES ('C004', 4, 'U004', 'frank@email.com', 'P004', 'The tour was unforgettable!');
```

```
INSERT INTO Comments (CommentsID, PostID, UserID, Email, ProfileID, CommentContent) VALUES ('C005', 5, 'U005', 'bon@email.com', 'P005', 'That live session was so good!');
```

-- Inserting data into Friends

```
INSERT INTO Friends (UserID, UserEmail, UserProfileID, FriendUserID, FriendUserEmail, FriendUserProfileID) VALUES ('U001', 'phoebe@email.com', 'P001', 'U002', 'taylor@email.com', 'P002');
```

```
INSERT INTO Friends (UserID, UserEmail, UserProfileID, FriendUserID, FriendUserEmail, FriendUserProfileID) VALUES ('U002', 'taylor@email.com', 'P002', 'U003', 'billie@email.com', 'P003');
```

```
INSERT INTO Friends (UserID, UserEmail, UserProfileID, FriendUserID, FriendUserEmail, FriendUserProfileID) VALUES ('U003', 'billie@email.com', 'P003', 'U004', 'frank@email.com', 'P004');
```

```
INSERT INTO Friends (UserID, UserEmail, UserProfileID, FriendUserID, FriendUserEmail, FriendUserProfileID) VALUES ('U004', 'frank@email.com', 'P004', 'U005', 'bon@email.com', 'P005');
```

```
INSERT INTO Friends (UserID, UserEmail, UserProfileID, FriendUserID, FriendUserEmail, FriendUserProfileID) VALUES ('U005', 'bon@email.com', 'P005', 'U001', 'phoebe@email.com', 'P001');
```